VISUAL C++.NET

WINDOWS FORM APPLICATION

Part One - Six

إنشاء

م.هيثم مالك فهمي محمد



محتويات الكتاب و الفهارس

Index Visual C++.Net

(1)Introduction to Microsoft Visual C++ .NET مقدمة في لغة السي المرائي

Closes	❌	Maximizes	🔲	Minimizes	🔽

شكل رقم (1-1)

بناء البرنامج يتم من خلال هذه المصطلحات التي يجب معرفتها لفهم أساسيات البناء للبرنامج VC++.Net و منها نتعلم ماذا تعني المقدمة في هذا البرنامج

graphical user interface (GUI) application	واجهة مستعمل بالرسوم
Win32 library	
graphical programming environment	بيئة البرمجة التخطيطية
The Integrated Development Environment	بيئة التنمية المتكاملة

البرمجة التخطيطية بَدأتْ أَنْ تَكُونَ مُعتَبَرة،في مايكروسوفت أصدرتْ منها لغة البرمجة vc++.net وهذا تابع لتقنية graphical user interface (GUI) application. المصمم بها نظام التشغيل Windows3.1 و هذه التقنية عبارة عن تحويل الأوامر لنظام doc إلي بيئة رسومي مرائية ، أما Win32 هو فقط مكتبة. مثل أكثر أَو كُلّ المكتبات العامة الأخرى، هو لَيسَ برمجة لغةِ. هو توثيق الذي يَسْمحُ للمبرمجين لمعرفةِ كيفَ هم يُمْكِنُ أَنْ يَستعملوا المكتبةَ لخَلْق التطبيقاتِ الذي يُمْكِنُ أَنْ يَجْري على نوافذ مايكروسوفتِ و عبارة graphical programming environment. فهي عبارة عن الحقيقة لجَعْل النوافذِ الذي يُبرمجُ سهل الوصول إلى كُلّ شخصِ، خَلقتْ مايكروسوفتَ فيشوال بيسك مايكروسوفتِ بيئة برمجة تخطيطيةِ حقيقيةِ. فيشوال بيسك أستعمل لغة مختلفة وأسهل، مسمّاة الأساسِ كمؤسسه. النتيجة الأولى كَانتْ ذلك فيشوال بيسكِ و أخيرا عبارة The Integrated Development Environment فهي تعني أيقونة النظامَ تُستَعملُ لتَمييز التطبيقِ بأنك تَستعملُ. تقريباً كُلّ تطبيق لَهُ أيقونة نظامُها الخاصةُ. تَحْملُ أيقونةُ النظامَ قائمتِها الخاصةِ للأعمالِ مثال لذالك انظر الرسم (1-1) . . .



إنشاء برنامج باستخدام VC++.net

أولا الدخول علي الشاشة الافتتاحية

1- قم بالضغط Click Leftعلي أيقونة التشغيل من علي سطح المكتب

2- انتظر حتى يتم تحميل البرنامج علي الحاسب انظر الشكل رقم (2-1)

3- الدخول علي الشاشة الافتتاحية.

Microsoft
Visual C++
2008

الشكل رقم (2-1)

ثانيا مكونات برنامج Microsoft Visual C++.Net

1- من الشاشة الافتتاحية سوف نجد عبارتين هامتين هما : انظر الشكل (2-2)

- Open و هذا لفتح برنامج قديم منشأ من قبل .
- Create و هذا لإنشاء برنامج جديد .

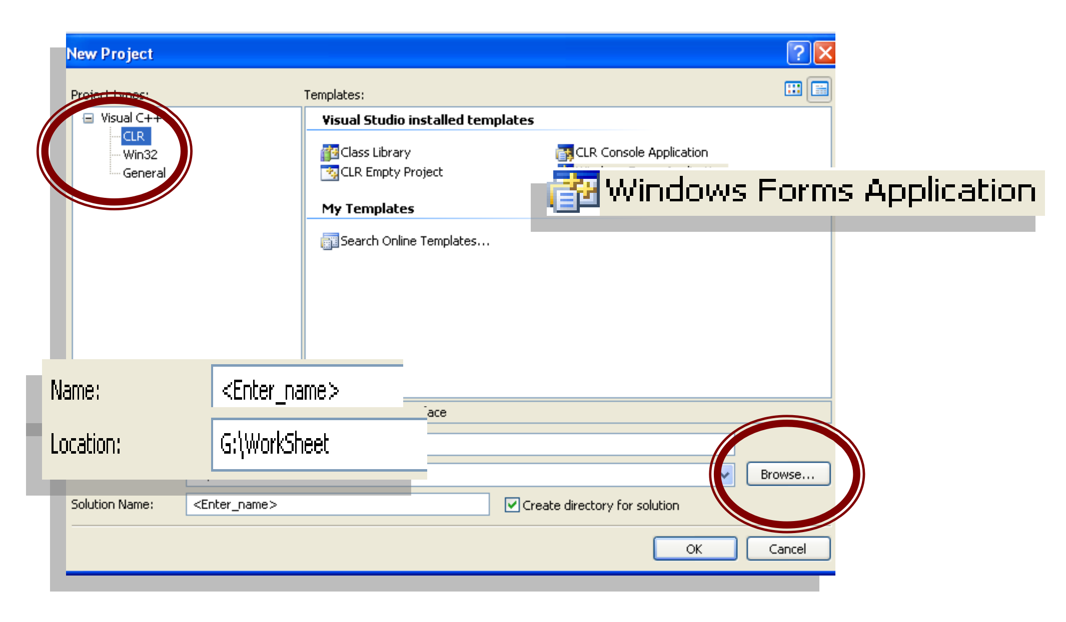
Open: Project...
Create: Project...

الشكل رقم (2-2)

2- لإنشاء البرنامج اضغط علي Project الموجودة بجوار Create

3- يظهر مربع حواري جديد و به مجموعه من الخيارات

- اختار من مربع الخيار project types CLR
- ثم من خانة Templates (Windows Froms Application)
- ثم من خانة Name أكتب اسم البرنامج المراد أنشاء
- ثم من خانة location اضغط علي زر Browse... لتحديد مكان تخزين البرنامج . .
- اضغط موافق OK لبدا العمل في البرنامج انظر الشكل (2-3).



الشكل رقم (2-3)

ثانيا التعرف علي مكونات برنامج Microsoft Visual C++.Net

مجموعة حفظ و الفتح في البرنامج

زر Play لتشغيل البرنامج المبرمج

Solution
و هي بها جميع أدوات الخاصة البرمجية و ما يخص ذلك البرنامج المنشي من class و خصائص.

اظهر صندوق الادوات

Properties
و هي التي تقوم بعمليات التغير في خصائص الاداوات و ظهورها علي Form.

أزرار إخراج الأحداث و نافذة الكود

البرنامج الأول

((رسالة ترحيب))

أولا : قم بإنشاء Form التالي :



الشكل رقم (3-1)

جدول يوضح كيفية استخدام Form				
م	اسم الأداة البرمجي	اسم الأداة للمستخدم	الحدث	الخاصية
1	Label1	من أنت	change	text
2	Button1	غريب	Click	text

جدول يوضح كيفية استخدام الخصائص			
م	اسم الأداة البرمجي	اسم الخاصية	قيمة الخيار
1	Label1	Auto size	false
2	Button1	غريب	

ثانيا : قم بعمل البرنامج كما يلي

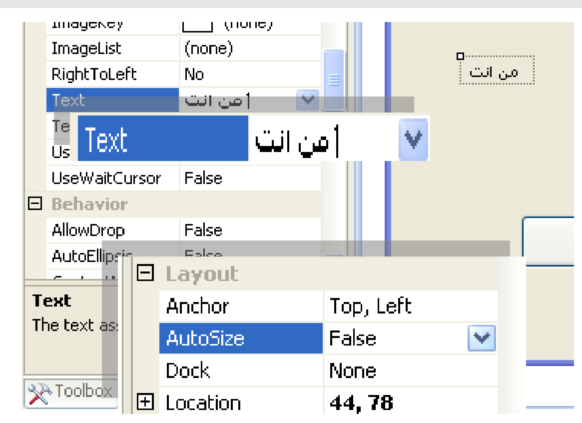
1- قم بإنشاء Label ثم من نافذة الخصائص قم بتغير الخاصية Text و الخاصية Auto size انظر الشكل 3-1 ، 3-2
2- قم بعمل زر Button ثم قم بتسميته غريب
3- اضغط علي الزر double-Click لدخول علي نافذة الكود و اكتب الكود التالي

```
#pragma endregion
	private: System::Void button1_Click(System::Object^  sender, System::EventArgs^  e) {

				this->label1->Text="مرحبا بكم في بلدكم الثاني مصر ";
			}
	};
}
```



الشكل رقم (3-3)



الشكل رقم (3-2)

4- و عند التشغيل و النقر علي زر " غريب " يعطي الناتج كما بالشكل (3-3)

الوصف و التعرف

عبارة this-> تعبر عن كل الأدوات و الخواص الموجودة داخل البرنامج نختار اسم الأداة label1 ثم نستخدم وسيط المعامل -> لاستخراج خصائص الأداة المستخدمة سوف استخدم الخاصية Text لأنها الخاصية الخاصة لوضع قيم و طباعتها أما عبارة = تستخدم مع الخصائص التي يرجا منها إعطاء نتائج ظاهرة و نهي البرنامج ب ;

تم البرنامج



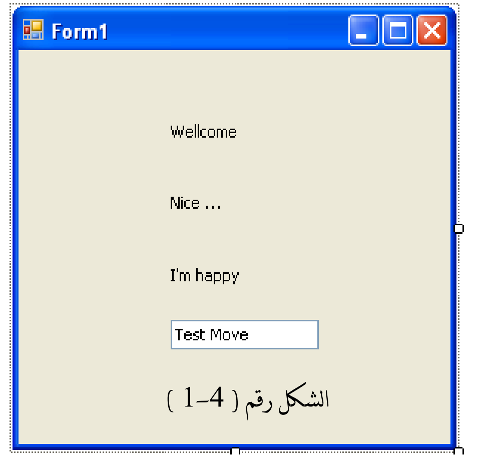
البرنامج الثاني

((الأحداث Events))

الحدث هي هدف يتحقق بوسائل مختلفة يمكن تحققه بواسطة النقر أو ضغطتين من الفارة أو باستخدام المرور علي كائن فيقرء محتويات الكود ثم ينفذها بمجرد وقوع الحدث .

البرنامج التالي يوضح كيف يمكن أن تتعرف علي الأحداث

كما هو في الشكل (4-1)



الشكل رقم (4-1)

جدول يوضح كيفية استخدام Form				
م	اسم الأداة البرمجي	اسم الأداة للمستخدم	الحدث	الخاصية
1	Label1	welcome	change	text
2	Label2	Nice	change	text
1	Label3	I'm happy	change	text
1	TextBox1	Test Move	change	text

البرنامج هو عبارة عن ثلاثة من Label و Textbox و هو يتم تغير الأحداث من Textbox و يغير في كل label علي حدا حسب الحدث المختار .

1- من الكائن textbox من علي form1 قم بتنشيط الكائن ثم اتجه إلي قائمة الخصائص و اختر منها زر الأحداث كما هو بالشكل (4- 2)

2- و بمجرد الضغط علي ذلك الزر يظهر نافذة الأحداث قم باختيار الحدث المراد و هو Mouse move

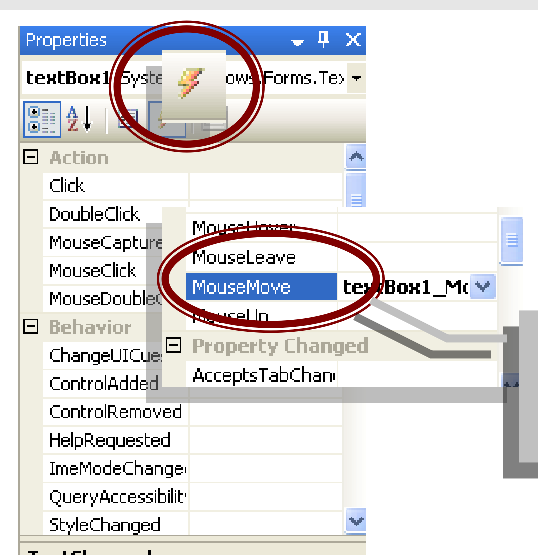
3- انقر double-click left علي الحدث و سوف يتم كتابة الكائن المراد عمل حدث له تلقائي ثم الدخول علي نافذة الكود

4- في الحدث الطبيعي للكائن textbox هو الحدث change و يكون كا التالي

```
private: System::Void textBox1_TextChanged(System::Object^  sender, System::EventArgs^  e) {
         }
```

أما بعد اختيارك للحدث Mouse Move يكون كالتالي

```
private: System::Void textBox1_MouseMove(System::Object^  sender,
System::Windows::Forms::MouseEventArgs^  e) {
}
```



انقر double-click left
علي الحدث و سوف يتم كتابة الكائن المراد عمل
حدث له تلقائي ثم الدخول علي نافذة الكود

الشكل رقم (4-2)

و بنفس الطريقة نختار الحدث ثم ندخل نافذة الكود ثم نكتب الكود داخل سطر كود الحدث . . فإذا وقع الحدث ينفذ ما بين السطور و هي الكود

أكتب أكواد البرنامج :

```
#pragma endregion
	private: System::Void textBox1_MouseMove(System::Object^  sender,
System::Windows::Forms::MouseEventArgs^  e) {

				 this->label1->AutoSize.FalseString;
				 this->label1->Text="You move in my Textbox nice ...";
			 }
	private: System::Void textBox1_Click(System::Object^  sender, System::EventArgs^  e) {

				 this->label2->Text="WellDone";

			 }
private: System::Void textBox1_DoubleClick(System::Object^  sender, System::EventArgs^  e) {

			 this->label3->Text="Play With Me";

		 }
private: System::Void textBox1_TextChanged(System::Object^  sender, System::EventArgs^  e) {
		 }

};
}
```

في هذا الكود سوف أقوم بتحويل الكائن label1 من مساحة محدودة إلى مساحة ثابتة كبيرة ...

أما الحدث **Formloud** يستخدم للواجهات فقط و هو فريد حيث كل كود يكتب في هذا الحدث ينفذ مباشرتا .

```
private: System::Void Form1_Load(System::Object^  sender, System::EventArgs^  e)
 {

 }
```

Form1

You move in my Textbox nice ...

WellDone

Play With Me

Test Move

بعد تشغيل البرنامج و تفعيل الأحداث

تم البرنامج



البرنامج الثالث

((الواجهات Forms))

الوجهات هي احدي تقنيات (GUI)

و سوف تعرف من خلال ذلك البرنامج علي طرق التعامل مع خصائص Form و التنقل بين الواجهات .

أولا إنشاء form1

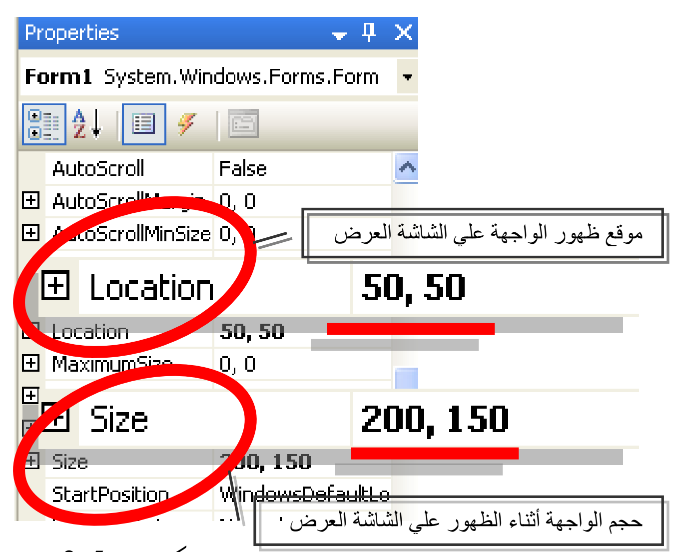
1- قم بإنشاء برنامج و قم بتسميته Formos ثم قم برسم الكائنات علي الواجهة كما هي في الشكل 5-1 .

2- من مربع الخصائص اجعل القيم كما بالشكل (5-2) .

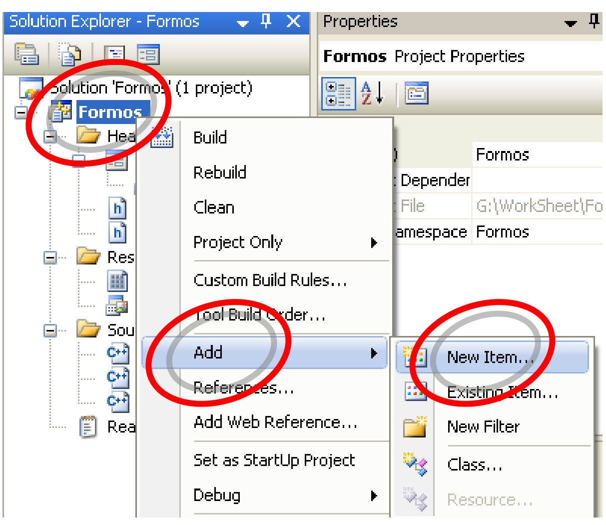
3- الخاصية location تجعلك تتحكم في مكان ظهور النافذة بعد تشغيل البرنامج

4- الخاصية sizeتتحكم في حجم النافذة من حيث الطول و العرض . . و سوف نستخدم الخاصيتين لجعل النافذتين form1 و form2 متطابقتين

الشكل رقم (5-1)



الشكل رقم (5-2)



الشكل رقم (5-3)

ثانيا انشاء form2

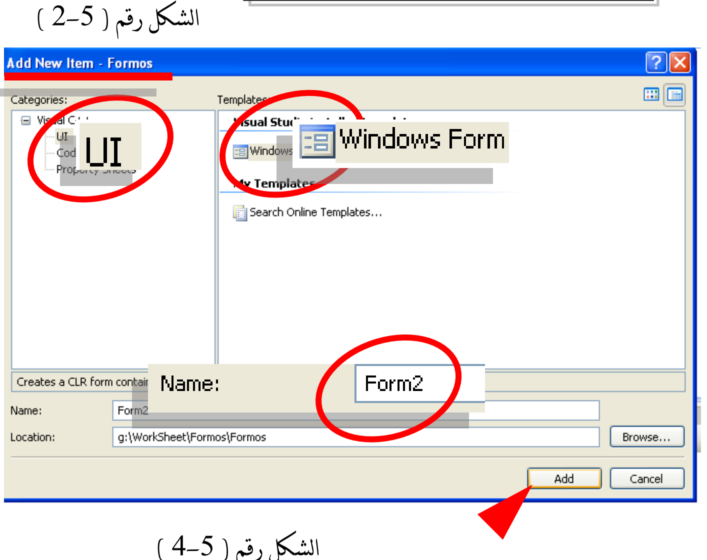
1- انظر إلي الشكل 5-3 وقم بعملها علي نافذة solution

2- بعد النقر علي الخيار New Item

3- يظهر مربع Add New Item

4- قم بكتابة اسم Form2 كما في الشكل 5-4

5- و فور انتهائك اضغط زر Add يتم بذلك عمل form2

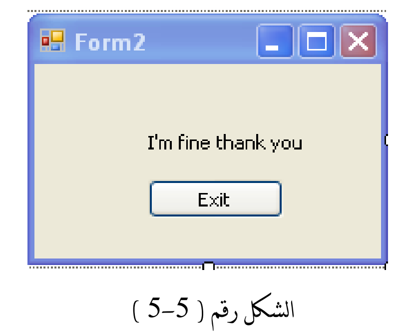


الشكل رقم (5-4)



ثالثا وضع الكائنات علي form2

1. قم بنشاء label و قم بتسميته I'm fine thank you
2. ثم قم بعمل زر Button و قم بتسميته Exit لعملية الخروج من البرنامج كما هو بالشكل 5-5
3. قم بضغط علي زر Exit لدخول علي نافذة الكود ثم أكتب كود الخروج من البرنامج



الشكل رقم (5-5)

```
private: System::Void button1_Click(System::Object^  sender, System::EventArgs^  e) {
                         Application::Exit();
                 }
```

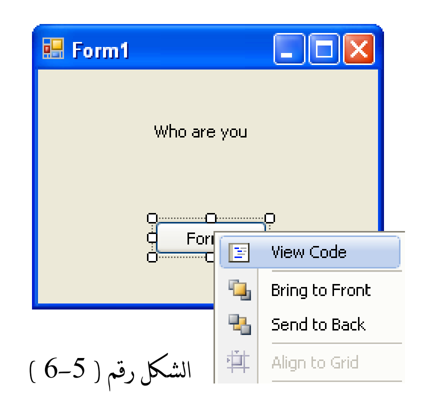
رابعا تعرف الكائن Form2 للبرنامج

1. اتجهة الي Form1 ثم ادخل علي نافذة الكود الخاصة بها ثم أكتب كود التعريف

و يكتب اعلي البرنامج أسفل عبارة #progma once

```
#pragma once
#include"Form2.h"
```

2. بعد ذلك اتجهة الي Form1 و اضغط علي زر المكتوب علية Form2 او انقر علية clickRight
3. ثم اختر الامر view code لدخول نافذة كود الزر كما هو موضح في الشكل 6-5
4. أكتب الكود التالي داخل الزر . . .



الشكل رقم (6-5)

```
#pragma endregion
      private: System::Void button1_Click(System::Object^  sender, System::EventArgs^  e) {

                         Form2^ form2=gcnew Form2();
                         form2->Show();
                 }
```

تم البرنامج

البرنامج الرابع

((أدوات التحميل Timer & ProgressBar))

1. قم بإنشاء Form ثم ضع بداخلها الكائن Timer و الكائن ProgressBar
2. و من نافذة الخصائص التابعة للكائن Timer قم بتغير القيمتين
   - Enabled = true
   - Interval = 500

و هذا يعني أن خاصية الإتاحة للكائن الزمني Timer فعال باختيار الخاصية true و ان العداد يساوي 500 وحدة زمنية

3. يجب ان تحدد نقطة البداية min و نقطه النهاية max من خاصية الكائن progressbar وذلك من خلال الخاصية
   - Min= 0
   - Max= 100
4. اضغط علي الكائن Timer و ادخل في نافذة الكود و قم بحجز متغيرات لعمل عداد زمني و يكون المتغير من النوع العام public
5. و يجب ان يكتب المتغير أسفل هذه عبارة

```
#pragma endregion
     public:
          int x;
```

6. قم بكتابة الكود التالي داخل الكائن Timer

```
private: System::Void timer1_Tick(System::Object^  sender, System::EventArgs^  e) {

                         x=x+1;
                         if(x<101)
                         {
                              this->progressBar1->Value=x;
                         }
           }
```

شرح الكود

سوف إعلان عن متغير من النوع العدد الصحيح كما هو موضح ثم بعد ذلك سوف اكتب الكود بتحميل المتغير x بقيمة x+1 و هذا يعني عمل عداد و ضع الجملة if بحد شرط و هو أذا كان قيمة المتغير x اكبر من 101 يتم توقف عملية العداد تسجيل الكائن progressbar و تحميله بقيمة العداد x

تم البرنامج



البرنامج الخامس

((الرسائل Message Box))

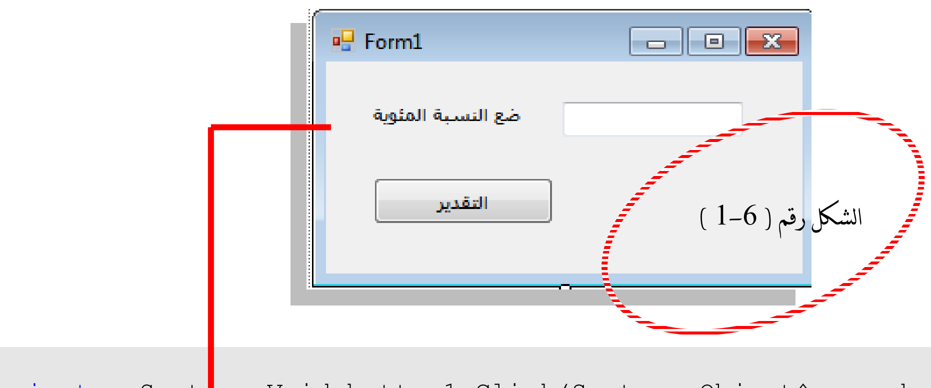
البرنامج يقوم بعمل مقارنات بين النسبة التي تكتب داخل Textbox الموجود داخل الفورم ثم يعطي الناتج في صورة Message Box و بيان التقدير نجاح أو رسوب

1- قم بإنشاء الفورم كما بالشكل (6-1)

2- اضغط علي زر Button 1 المسمي " التقدير" نقرتين مزدوجة لدخول إلي نافذة الكود ثم أكتب الكود التالي

3- أما عبارة if هي العبارة الشرطية التي تحدد الشرط و عبارة else هي في حالة عدم تحقق الشرط الأول يحقق الشرط الثاني

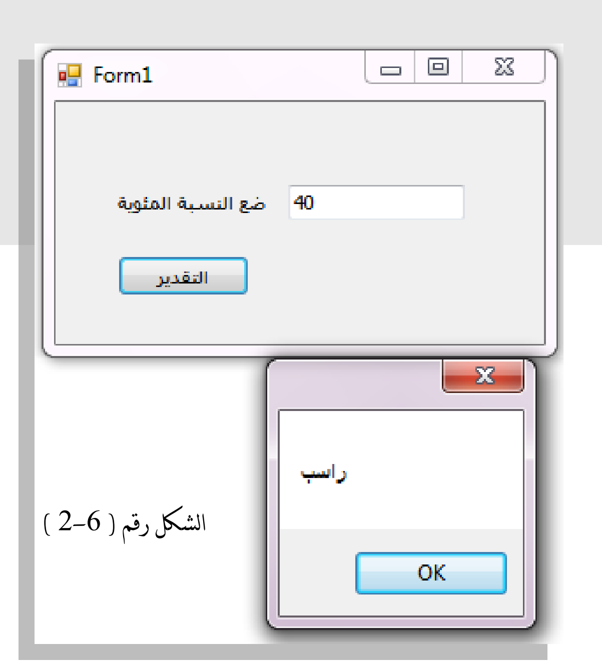
4- الشكل 6-2 يظهر الناتج بعد إتمام البرنامج



الشكل رقم (6-1)

```
private: System::Void button1_Click(System::Object^  sender, System::EventArgs^  e) {
                 double present=System::Convert::ToDouble(textBox1->Text);
                 if (present>=50)
                 {
                       MessageBox::Show("نـاجح");
                 }
                 else
                 {
                       MessageBox::Show("راسب");

                 };
 }
```



الشكل رقم (6-2)

تم البرنامج



البرنامج السادس

((التغير في كلمات علي زر تفعيل Change Button))

البرنامج عبارة عن زر يغير الكلمة الموجودة بداخلة إلي كلمة أخري . .

1- قم بعمل زر ثم اضغط علية Click Right و اختار view code

2- اتجه إلي الكود المحفوظ داخل البرنامج تلقائي و هو . .

```
// button1
//
this->button1->Location = System::Drawing::Point(106, 103);
this->button1->Name = L"button1";
this->button1->Size = System::Drawing::Size(75, 23);
this->button1->TabIndex = 0;
this->button1->Text = L"Welcome";
this->button1->UseVisualStyleBackColor = true;
//
```

يمكن من خلال هذه العبارة تغير اسم الزر دون الرجوع إلي نافذة الخصائص و تغير قيمة الخاصية text

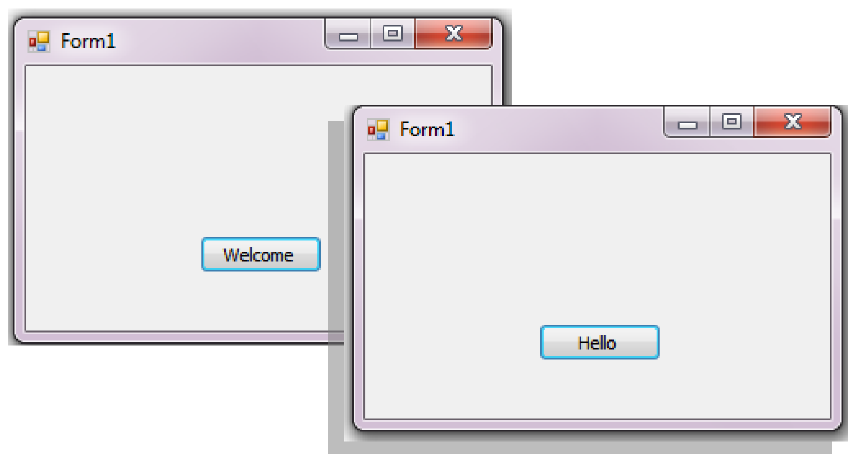
و هنا تحدد صفات الكائن من اسم البرمجي و الاسم الحالي للمستخدم و تجد أمام كل كلمة العبارة التالية L"button1" او L"Welcome"

3- اذهب إلي نافذة الفورم و اضغط علي زر Button1 الذي يحمل عبارة Welcome وانقر علية مرتين لدخول علي نافذة الكود

4- ثم اكتب الكود الذي يقوم بعملية تغير الكلمة Welcome إلي عبارة Hello

```
private: System::Void button1_Click(System::Object^  sender, System::EventArgs^  e) {

                 if(this->button1->Text=="Welcome")
                 {
                        this->button1->Text=L"Hello";
                 }
                 else
                 {
                        this->button1->Text=L"Welcome";
                 };
```



تم البرنامج

البرنامج السابع

((إظهار صورة علي نافذة))

يقوم البرنامج بإظهار صورة علي الفورم بمجرد الضغط علي زر " صورة " و هذه الصورة مربوطة بمسار محدد

1- قم بإنشاء زر Button1 و قم بتسميته " صورة "
2- وإنزال الكائن pictureBox1 علي الفورم انظر الشكل 7-1
3- ثم قم بالنقر علي زر " صورة " نقر مزدوج لدخول علي نافذة الكود
4- و بمجرد الدخول علي نافذة الكود اكتب الكود التالي

و هذا الكود يوضح كيفية نقل المسار الموجود للصورة إلي نافذة الفورم

```
private: System::Void button1_Click(System::Object^  sender, System::EventArgs^  e) {
         this->pictureBox1->ImageLocation=L"F:\\WorkSheet\\Tutorial Exam VC++\\pic\\x.GIF";
                         }
```

نجد الخاصية Imagelocation هي المسئولة عن عملية الكشف عن مسار الصور داخل نظام التشغيل و ربطها بالبرنامج

Form1
65
صورة

الشكل رقم (6-2)

تم البرنامج

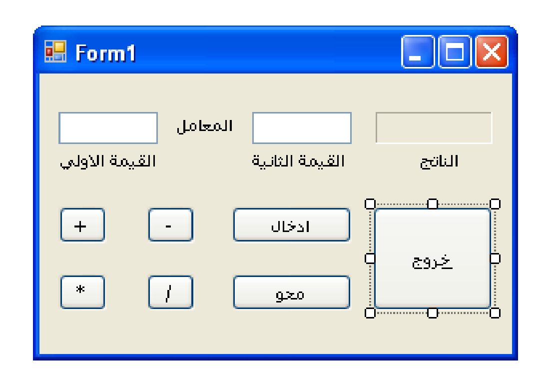
البرنامج الثامن

((الحاسبة))

برنامج الحاسبة و المعاملات

أولا : قم بإنشاء Form التالي :

جدول يوضح كيفية استخدام Form				
م	اسم الأداة البرمجي	اسم الأداة للمستخدم	الحدث	الخاصية
1	Label1	القيمة الأولي	change	text
2	Label2	القيمة التالية	change	text
3	Label3	الناتج	change	text
4	Label4	فارغ	change	
5	Label5	المعامل	change	text
6	Button1	+	Click	text
7	Button2	-	Click	text
8	Button3	*	Click	text
9	Button4	/	Click	text
10	Button5	إدخال	Click	text
11	Button6	محو	Click	text
12	Button7	خروج	Click	text
13	Textbox1	فارغ	change	
14	Textbox2	فارغ	change	



قم بضغط علي زر + أي Button1 و اكتب الكود التالي في نافذة الكود و لكن قبل كتابة الكود داخل الزر يجب تعريف المتغيرات التي تتحكم في سير البرنامج و هما :-

شرح المتغيرات

1. سوف إعلان عن متغير من النوع cher و هو c لكي يحمل إشارات العمليات الحسابية من طرح و جمع و غيرها
2. و أيضا سوف إعلان عن المتغير x,y و هما من النوع double و هذا لحمل القيم التي سوف توضع في textbox1, textbox2
3. أما المتغير d فسوف يقوم بعملية إظهار الناتج الحسابي

و تكتب كود المتغيرات أسفل عبارة # pragma endregion

```
#pragma endregion
char c ;
double x,y,d;
```

1- ثم اكتب الكود التالي داخل زر + أي Button1 و هو :

```
private: System::Void button1_Click(System::Object^  sender, System::EventArgs^  e) {
                 this->label5->Text="+";
                 c='+';
         }
```

وهو عبارة عن و ضع العلامة + بمجرد الضغط علي زر Button 1 و يغير كلمة المعامل الي + ثم تخزين العلامة + داخل المتغير c من النوع الحرفي

2- اضغط علي زر button2, button3,button4 و اكتب بهم المعاملات الحسابية المتبقية و هي -و*و/ ثم اكتب الاكواد الخاصة بهم :

```
private: System::Void button2_Click(System::Object^  sender, System::EventArgs^  e) {

                 this->label5->Text="-";
                 c='-';
         }
```

```
private: System::Void button3_Click(System::Object^  sender, System::EventArgs^  e) {

                 this->label5->Text="*";
                 c='*';
         }
```

```
private: System::Void button4_Click(System::Object^  sender, System::EventArgs^  e) {

                 this->label5->Text="/";
                 c='/';
         }
```

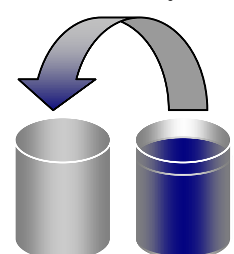
نجد المعامل '/' أو '*'
و هذا يعني أن أي معامل بين هذان الشرطتين 'ما بينهما' يعامل كنه معامل أساسي و ليس نص

3- أما زر button5 المسمي زر الإدخال يقوم بحساب القيمتان التي بين textbox2 , textbox1 و المعامل الحسابي المختار

شرح الكود:

4- الكود رقم 1 هو عبارة عن متغير x يوضع بداخلة قيمة محتوى textbox 1 و ذلك عن طريق استخدام الأمر System أي مكتبة أوامر النظام ثم اختيار convert معامل التحويل ثم اختيار معامل العدد toint32 و بذالك يتم و ضع القيمة المتواجدة في textbox 1 داخل المتغير x **حسب قاعدة البرمجية المليان يصب علي الفارغ ...**

5- أما الكود رقم 2 هو عملية عكس الكود السابق و هو وضع قيمة المتغير الذي تم عمل المعاملات الحسابية له إلي Label4 و باستخدام نفس القاعدة ..






```
private: System::Void button5_Click(System::Object^  sender, System::EventArgs^  e) {

         x=System::Convert::ToInt32(this->textBox1->Text);
          y=System::Convert::ToInt32(this->textBox2->Text);

          switch (c)
          {
          case '+': d=x+y;
                break;
                case '-': d=x-y;
                break;
                case '*': d=x*y;
                break;
                case '/': d=x/y;
                break;

                this->label4->Text=System::Convert::ToString(d);

          }

      }
```

1
2

6- زر رقم 6 و هو زر المحو القيم الموجودة علي Textbox 1 , textbox 2 , label4 وذلك باستخدام علامة "" مع الخاصية text .

و هو التالي :

```
private: System::Void button6_Click(System::Object^  sender, System::EventArgs^  e) {

         this->textBox1->Text="";
          this->textBox2->Text="";
          this->label4->Text="";
          this->label5->Text="";

      }
```

7- زر رقم 7 زر الخروج

```
private: System::Void button7_Click(System::Object^  sender, System::EventArgs^  e) {

         Application::Exit();
      }
```

تم البرنامج

البرنامج التاسع

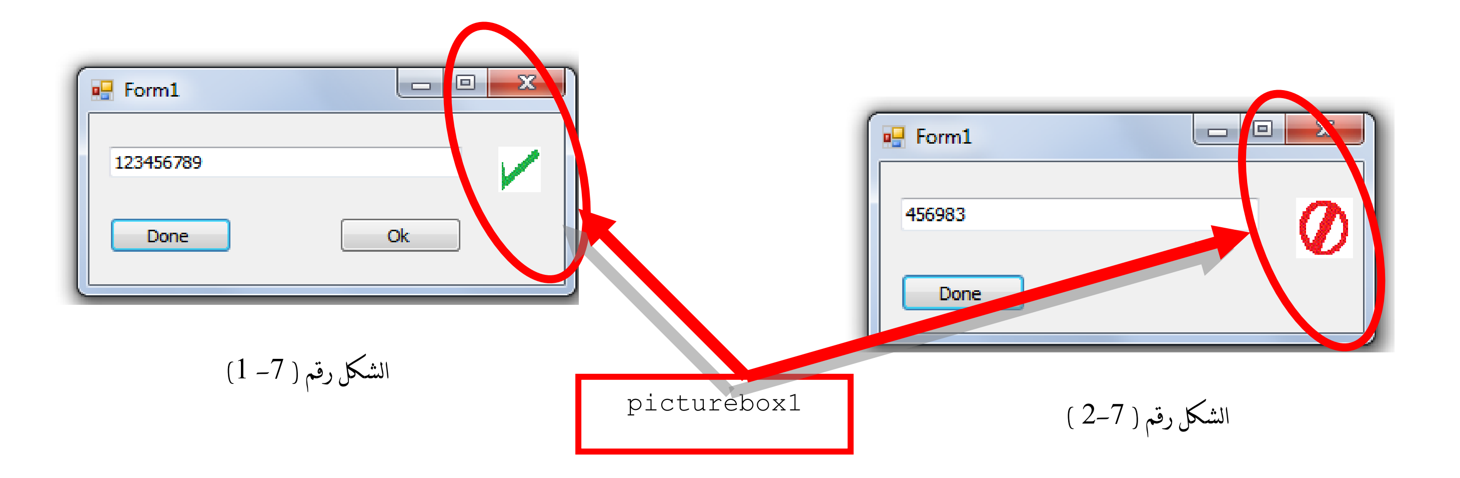
((كلمة المرور))

شرح البرنامج

البرنامج عبارة عن كلمة مرور تكتب داخل textbox و بمجرد الضغط على زر Done يحمل صورة علامة صح الموجودة بالشكل 7 - 1 مع ظهور الزر المخفي Ok و هذا في حالة كلمة المرور الصحيحة

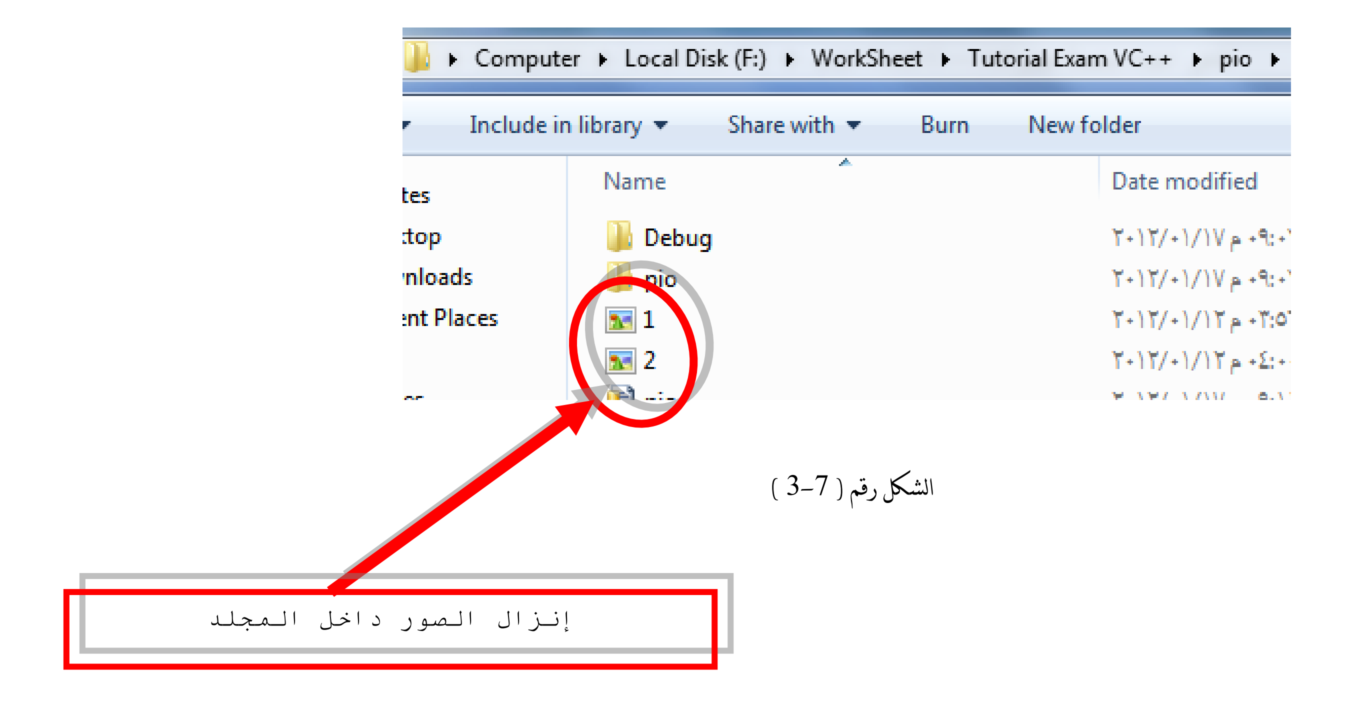
أما في حالة كلمة المرور الغير صحيحة يحمل صورة التوقف عن العمل كما بالشكل 7 - 2

ضع داخل مجلد البرنامج الصور 1.GIF 2.GIF لحين الاستدعاء بمسار المحدد لهم كما بالشكل 7- 3



الشكل رقم (7- 1)

الشكل رقم (7-2)



الشكل رقم (7-3)

ثم اكتب الكود التالي داخل زر Done أي button1

```
     int x ;
     private: System::Void button1_Click(System::Object^  sender, System::EventArgs^  e) {

       x=System::Convert::ToInt32(this->textBox1->Text);
      if (x==123456789)
            {
      this->pictureBox1->ImageLocation="F:\\WorkSheet\\Tutorial Exam VC++\\pio\\1.GIF";
            this->button2->Visible="True";
      }

      else {
      this->pictureBox1->ImageLocation="F:\\WorkSheet\\Tutorial Exam VC++\\pio\\2.GIF";
             };
```

شرح الكود

1. سوف أقوم بعملية تعريف للمتغير x بأنه من النوع الصحيح int
2. داخل زر button1 سوف أقوم بأخذ محتوى القيمة العددية التي سوف تكتب داخل Textbox1 و دخولها إلي المتغير x
3. سوف استخدم قاعدة if الشرطية في حالة كتابة القيمة داخل textbox سوف تكون المقارنة بين الرقم الموجود و الرقم الموضوع داخل المربع النصي
4. في حالة تطابق الرقمان سوف يكون الناتج إظهار الصورة في مربع picturebox1 وهي 1.GIF مع ظهور button2 من صورته المخفية إلي الصورة المرئية باستخدام الكود

```
this->button2->Visible="True";
```

5. أما في حالة عدم تطابق الشرط يستخدم العبارة else { }; و يكتب بين القوسين كود ظهور الصورة في حالة الخيار الخاطئ .

تم البرنامج

البرنامج العاشر

((الزمن و السرعة))

البرنامج عبارة عن قيمة تحول إلي زمن أو سرعة بواسطة الكائن radiobutton هي اختيار الزمن مرة و اختيار السرعة .

أولا : قم بإنشاء Form التالي :

جدول يوضح كيفية استخدام Form

م	اسم الأداة البرمجي	اسم الأداة للمستخدم	الحدث	الفورم
1	Label1	القيمة	change	
2	Label2	الناتج	change	
3	Button1	التحويل	Click	
4	Textbox1	فارغ	change	
5	Textbox2	فارغ	change	
6	radioButton1	الزمن	CheckedChanged	
7	radioButton2	السرعة	CheckedChanged	

ثانيا : بناء البرنامج بالا كواد :

```
#pragma endregion
          int y;
          int x;
          int z;

     private: System::Void button1_Click(System::Object^  sender, System::EventArgs^  e) {
                    if(this->radioButton1->Checked==false){


                         y=System::Convert::ToInt32(this->textBox1->Text);
                         z=y*100;
                         this->textBox2->Text=System::Convert::ToString(z);
                    }
                    else{
                         y=System::Convert::ToInt32(this->textBox1->Text);
                         x=y*60;
                         this->textBox2->Text=System::Convert::ToString(x);

                    };
```

1

ثالثا : تعريف المتغيرات y,x,z

1. Y يقوم هذا المتغير بأخذ القيمة الموجودة في TextBox1 إلي مخزن لحين استرجاعها
2. X يقوم بعملية الحسابية و وضعها باخله x=y*60; ثم عرضها علي textbox2 أي الناتج
3. Z يقوم بعملية الحسابية و وضعها باخله z=y*100; ثم عرضها علي textbox2 أي الناتج

رابعا : قاعدة if

و هي الاحتمال بين شرطين هما تنشيط radioButton1 أو عدم التنشيط في حالة عدم التنشيط يتم تنفيذ الكود رقم و احد إلي قوس الإغلاق و الكود عبارة عن تنفيذ عملية حسابية ثم طباعتها في textbox2 و تحميل قيمة المتغير z بقيمته .

تم البرنامج

اللهم أسالك علم نافع و جسدا علي البلاء صابرا و الهدي و التقي و مصاحبة المصطفي من قول و عمل و صلي الله علي نبينا محمد يا أيها الذين امنوا صلوا علية و سلموا تسليما...

لتواصل
جمهورية مصر العربية
أسوان
هيثم مالك فهمي محمد عبد الرحيم المزرقاني الشريف
Hima_1700@hotmail.com
01067107750

